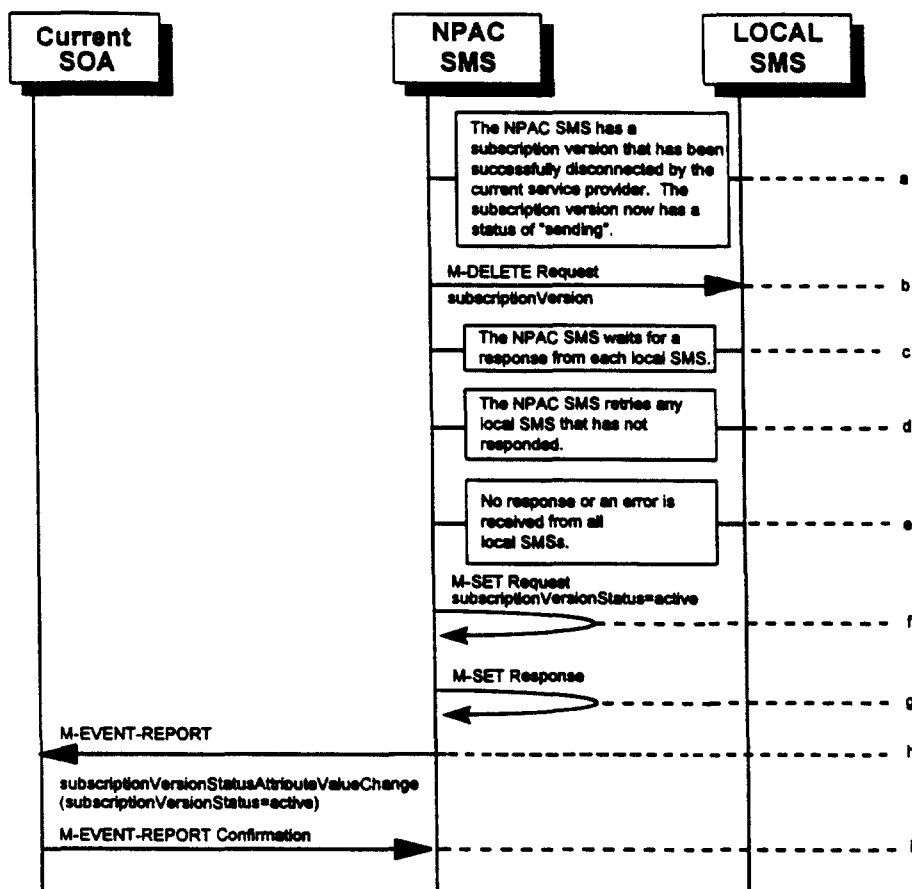### 6.5.4.3. SubscriptionVersion Disconnect: Failure to Local SMS

This scenario shows the broadcast of a disconnected subscription that fails to all of the Local SMSs.
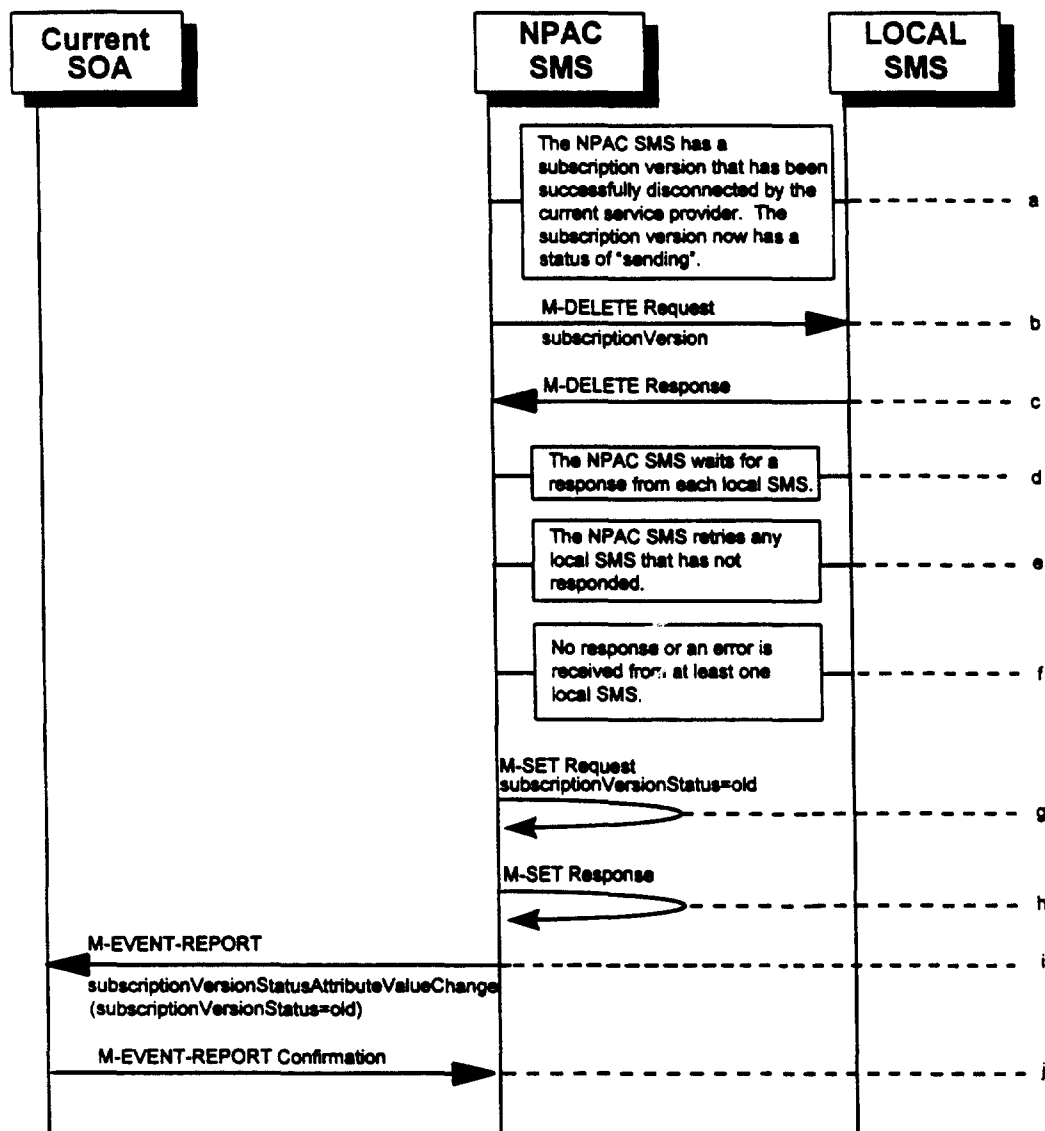

a. The NPAC SMS has an active subscription version that has been successfully disconnected by the current service provider using the subscriptionVersionDisconnect action. The subscription version now has a status of "sending".

b. NPAC SMS issues the M-DELETE to all Local SMSs for the subscriptionVersion.

c. NPAC SMS waits for a response from each Local SMS.

d. NPAC SMS retries any Local SMS that has not responded.

e. No response or an error is received from all Local SMSs.

f. NPAC SMS issues the M-SET to update the current subscriptionVersionNPAC object's subscriptionVersionStatus to "active" from "sending". It will also update the subscriptionFailed-SP-List with the service provider ID and name of all the Local SMSs.

g. NPAC SMS responds to the M-SET.

h. NPAC SMS sends the subscriptionVersionStatusAttributeValueChange M-EVENT-REPORT to the current service provider SOA with the current status and failedSP-List.

i.   Current service provider SOA issues the M-EVENT-REPORT confirmation.

6.5.4.4. SubscriptionVersion Disconnect: Partial Failure to Local SMS

This scenario shows the broadcast of a disconnected subscription that fails to one or more, but not all, of the Local SMSs.


a. The NPAC SMS has an active subscription version that has been successfully disconnected by the current service provider using the subscriptionVersionDisconnect action. The subscription version now has a status of "sending".

b. NPAC SMS issues the M-DELETE to all Local SMSs for the subscriptionVersion.

c. Local SMSs should respond successfully to the M-DELETE.

d. NPAC SMS waits for a response from each Local SMS.

e. NPAC SMS retries any Local SMS that has not responded.

f. No response or an error is received from at least one Local SMS.

g. NPAC SMS issues the M-SET to update the current subscriptionVersionNPAC object's subscriptionVersionStatus to "old" from "sending". It will also update the subscriptionFailed-SP-List with the service provider ID and name of the Local SMSs that failed to successfully receive the broadcast.

h. NPAC SMS responds to the M-SET.

i. NPAC SMS sends the subscriptionVersionStatusAttributeValueChange M-EVENT-REPORT to the current service provider SOA with the current status and failedSP-List.

j. Current service provider SOA issues the M-EVENT-REPORT confirmation.

6.5.4.5. Subscription Version Disconnect: Resend Successful to Local SMS

This scenario shows a successful resend of a disconnect for a subscription that fails to one or more of the Local SMSs. The resend of a failed disconnect can only be performed by authorized NPAC personnel.
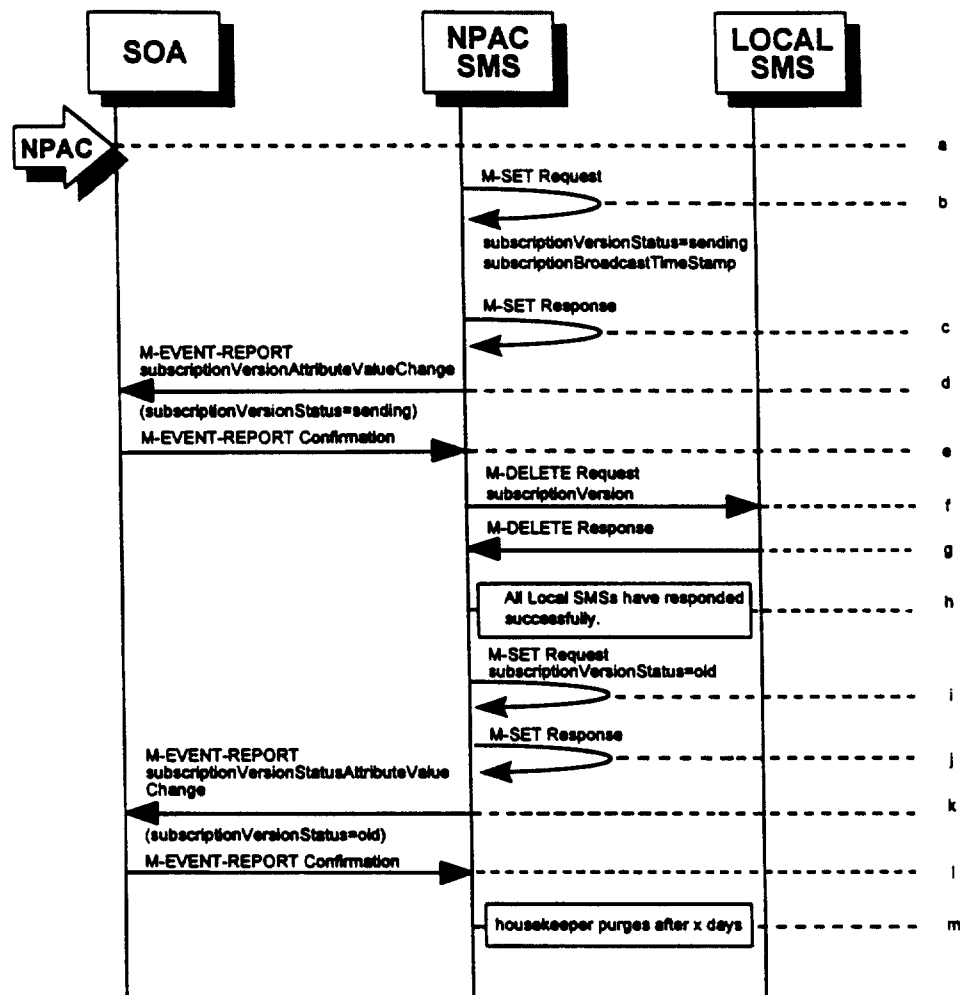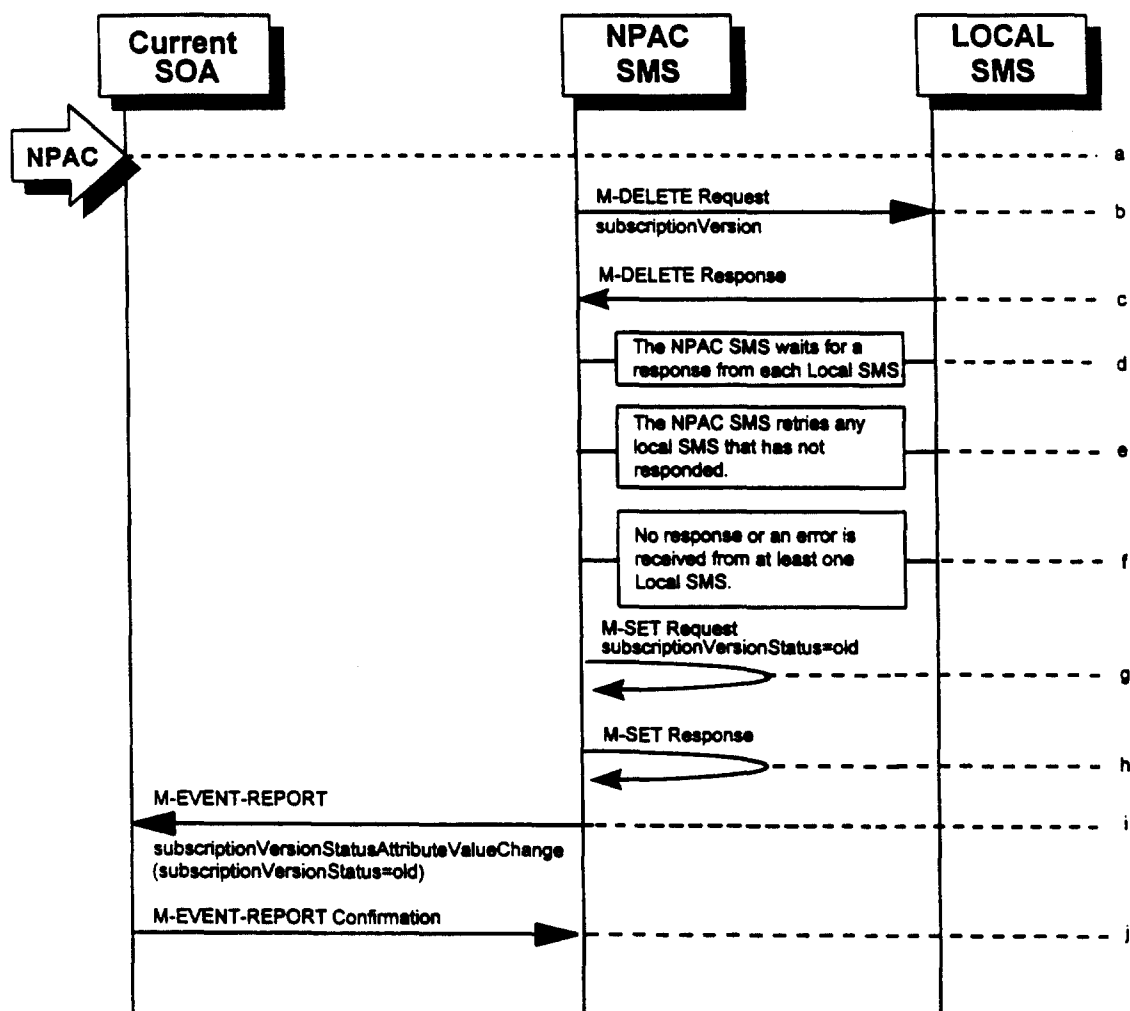

a. NPAC personnel take action to resend a failed disconnect for a subscription version.

b. NPAC SMS issues an M-SET to the existing subscriptionVersionNPAC object to set the status to "sending".

c. NPAC SMS responds to whether M-SET was successful.

d. NPAC SMS notifies service provider SOA of status change to "sending."

e. Service provider SOA confirms event report.

f. NPAC SMS sends out an M-DELETE on the subscriptionVersion to all previously failed Local SMSs.

g. Each Local SMS responds with a successful M-DELETE reply.

h. All Local SMSs respond successfully.

i. NPAC SMS issues M-SET updating the subscriptionVersionStatus to old for subscriptionVersionNPAC objects. It also sets the subscriptionModifiedTimeStamp and subscriptionDisconnectCompleteTimeStamp.

j. NPAC SMS responds to M-SET.

k. NPAC SMS issues an M-EVENT-REPORT for the subscriptionVersionStatus equal to "old."

l. Service provider SOA responds to M-EVENT-REPORT.

m. After a tunable amount of days, the subscription version is purged by the NPAC SMS housekeeping process.

6.5.4.6. Subscription Version Disconnect: Resend Failure to Local SMS

This scenario shows a failure on a resend of a subscription disconnect that failed previously to one or more of the Local SMSs. The resend of a failed disconnect for a subscription can only be performed by authorized NPAC personnel.

a. NPAC personnel take action to resend a failed disconnect for a subscription version.

b. NPAC SMS issues the M-DELETE to all Local SMSs for which to the disconnect previously failed for the subscriptionVersion.

c. Local SMSs should respond successfully to the M-DELETE.

d. NPAC SMS waits for a response from each Local SMS.

e. NPAC SMS retries any Local SMS that has not responded.

f. No response or an error is received from at least one or all Local SMSs.

g. NPAC SMS issues the M-SET to update the current subscriptionVersionNPAC object's subscriptionVersionStatus to "old" or "active" (if all Local SMSs failed) from "sending". It will also update the

subscriptionFailed-SP-List with the service provider ID and name of the Local SMSs that failed to successfully receive the broadcast.

h. NPAC SMS responds to the M-SET.

i. NPAC SMS sends the subscriptionVersionStatusAttributeValueChange M-EVENT-REPORT to the current service provider SOA with the current status and failedSP-List.

j. Current service provider SOA issues the M-EVENT-REPORT confirmation.

6.5.5. Conflict Scenarios

A situation has arisen which causes the NPAC SMS or NPAC personnel to place the subscriptionVersion into conflict.

A subscription version can be removed from conflict by the NPAC personnel or the new service provider SOA.

6.5.5.1. SubscriptionVersion Conflict and Conflict Resolution by the NPAC SMS

This scenario shows a version being placed into conflict and removed from conflict by the NPAC personnel.
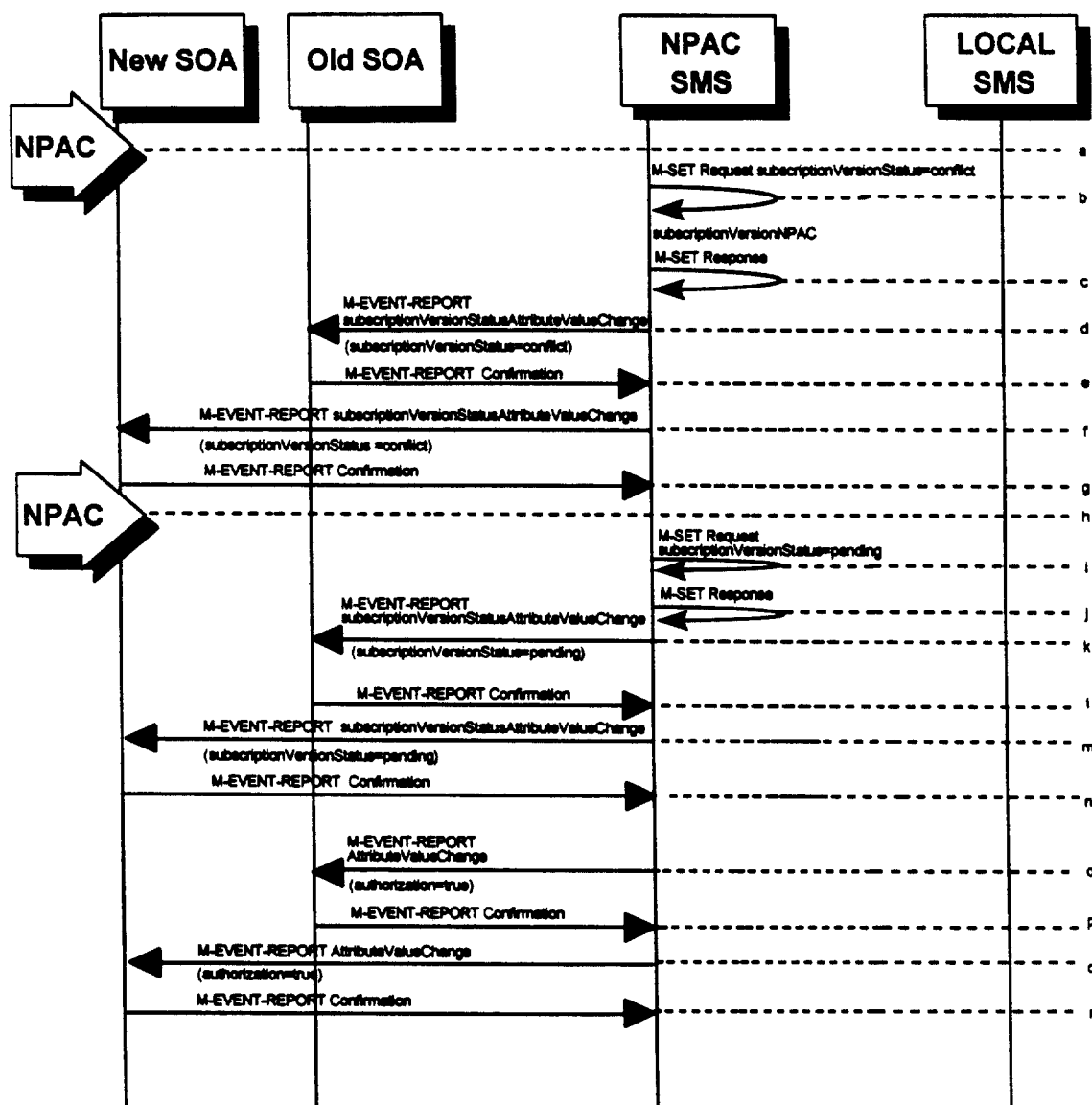
a. NPAC personnel or NPAC SMS take action to set the status of a subscription to "conflict."

b. NPAC SMS issues M-SET request to update subscriptionVersionStatus to "conflict," subscriptionConflictTimeStamp, and subscriptionModifiedTimeStamp in the subscriptionVersionNPAC object.

c. NPAC SMS issues an M-SET response. If the M-SET fails, processing for this scenario stops.

d. NPAC SMS issues an M-EVENT-REPORT subscriptionVersionStatusAttributeValueChange to old service provider SOA.

e. The old service provider SOA returns an M-EVENT-REPORT confirmation to the NPAC SMS.

f. NPAC SMS issues subscriptionVersionStatusAttributeValueChange for status to new service provider SOA.

g. The new service provider SOA returns an M-EVENT-REPORT confirmation to the NPAC SMS.

h. Once the conflict is resolved, NPAC personnel take action to remove the subscriptionVersion from conflict.

i. NPAC SMS issues an M-SET request to update the subscriptionModifiedTimeStamp and the subscriptionVersionStatus to "pending."

j. NPAC SMS issues an M-SET response. If the M-SET fails, processing for this scenario stops.

k. NPAC SMS issues subscriptionVersionStatusAttributeValueChange for the new status to the old service provider SOA.

l. The old service provider SOA returns an M-EVENT-REPORT confirmation to the NPAC SMS.

m. NPAC SMS issues subscriptionVersionStatusAttributeValueChange for the new status to the new service provider SOA.

n. The new service provider SOA returns an M-EVENT-REPORT confirmation to the NPAC SMS.

o. NPAC SMS sends a subscriptionVersionStatusAttributeValueChange to set the old service provider's authorization to "TRUE".

p. The old service provider SOA returns an M-EVENT-REPORT confirmation to the NPAC SMS.

q. NPAC SMS sends an AttributeValueChange to set the new service provider authorization to "TRUE".

r. The new service provider SOA returns an M-EVENT-REPORT confirmation to the NPAC SMS.

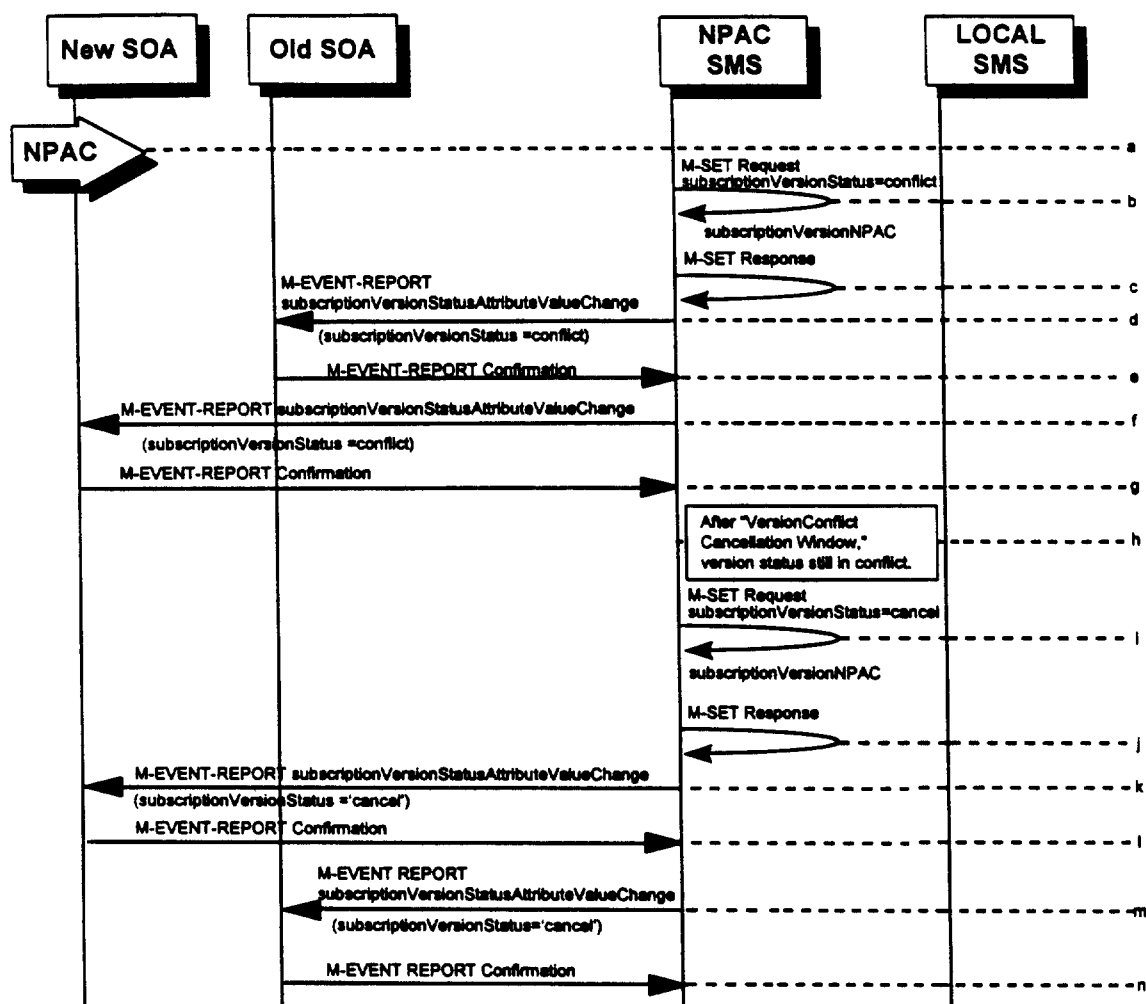6.5.5.2. Subscription Version Conflict Removal by the New Service Provider SOA

In this scenario, the new service provider elects to remove the subscription version from conflict.
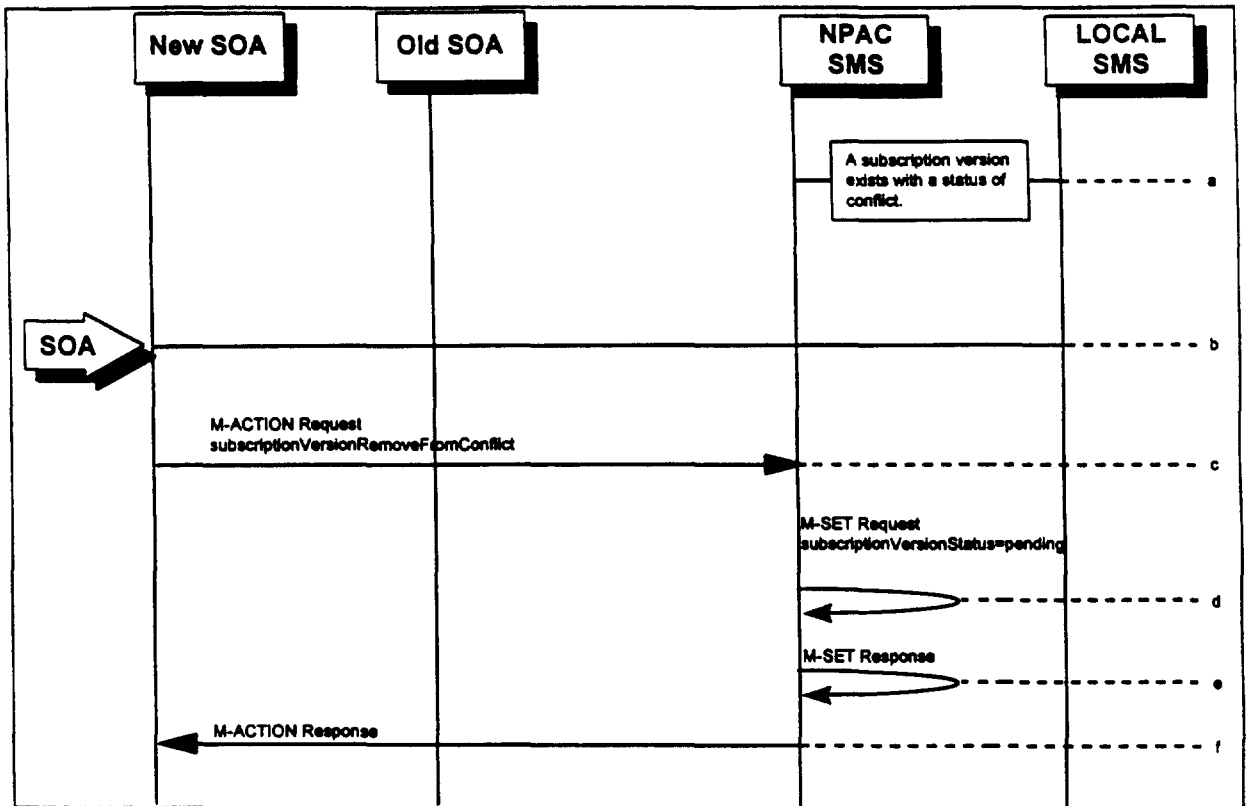
a. A subscription version exists on the NPAC SMS with a status of conflict.

b. The new service provider SOA personnel take action to remove the subscription version from conflict.

c. The new service provider SOA sends the M-ACTION subscriptionVersionNewSP-RemoveFromConflict specifying the subscription_-version_-TN or~~and~~ subscription version ID of the subscription version in conflict.

d. If the request is valid, the NPAC SMS will set the status to "pending".
    The request will be denied and an error returned if the subscriptionOldSP-Authorization was set to conflict by the old service provider and the conflict restriction window has not expired.

e. The NPAC SMS responds to its own M-SET.

f. The NPAC SMS responds to the M-ACTION with success or failure and reason for failure.
    The processing now continues with the subscriptionVersionStatusAttributeValueChange notices going out to the new and old service provider SOAs.

6.5.5.3. SubscriptionVersion Conflict: No Conflict Resolution

This scenario shows the action taken at the NPAC SMS when service providers do not reach a conflict resolution.

a. NPAC personnel or NPAC SMS take action to set a subscriptionVersionStatus to "conflict."

b. NPAC SMS issues an M-SET request to set the subscriptionVersionStatus to "conflict," the subscriptionConflictTimeStamp, and the subscriptionModifiedTimeStamp in the subscriptionVersionNPAC object.

c. NPAC SMS responds to M-SET. If the M-SET fails, processing stops for this scenario until the M-SET completes successfully.

d. NPAC SMS issues subscriptionVersionStatusAttributeValueChange to old service provider SOA for the new "conflict" status.

e. The old service provider SOA returns an M-EVENT-REPORT confirmation to the NPAC SMS.

f. NPAC SMS issues subscriptionVersionStatusAttributeValueChange to new service provider SOA for the "conflict" status.

g. The new service provider SOA returns an M-EVENT-REPORT confirmation to the NPAC SMS.

h. "Version Conflict Cancellation Window" expires without conflict resolution.

i. NPAC SMS issues an M-SET request to set the subscriptionVersionStatus to "cancel" in the subscriptionVersionNPAC object and sets the subscriptionCancellationTimeStamp and subscriptionModifiedTimeStamp.

j. NPAC SMS responds to M-SET. If the M-SET fails, processing stops for this scenario until the M-SET is successfully completed.

k. NPAC SMS issues attribute value change for status to new service provider SOA for the "cancel" status.

l. The new service provider SOA returns an M-EVENT-REPORT confirmation to the NPAC SMS.

m. NPAC SMS issues attribute value change for status to old service provider SOA for the "cancel" status.

n. The old service provider SOA returns an M-EVENT-REPORT confirmation to the NPAC SMS.

6.5.5.4. Subscription Version Conflict by Old Service Provider Explicitly Not Authorizing (First Create)

The old service provider SOA can put a pending subscription version into conflict by setting its authorization flag to off. This can be done on the subscriptionVersionOldSP-Create action, subscriptionVersionModify action, or M-SET of the attribute on the subscription version object.

This scenario shows the old service provider putting a new pending subscription version into conflict by turning the authorization flag off on the subscriptionVersionOldSP-Create. In this case, the old service provider's create action is the first sent to the NPAC SMS.
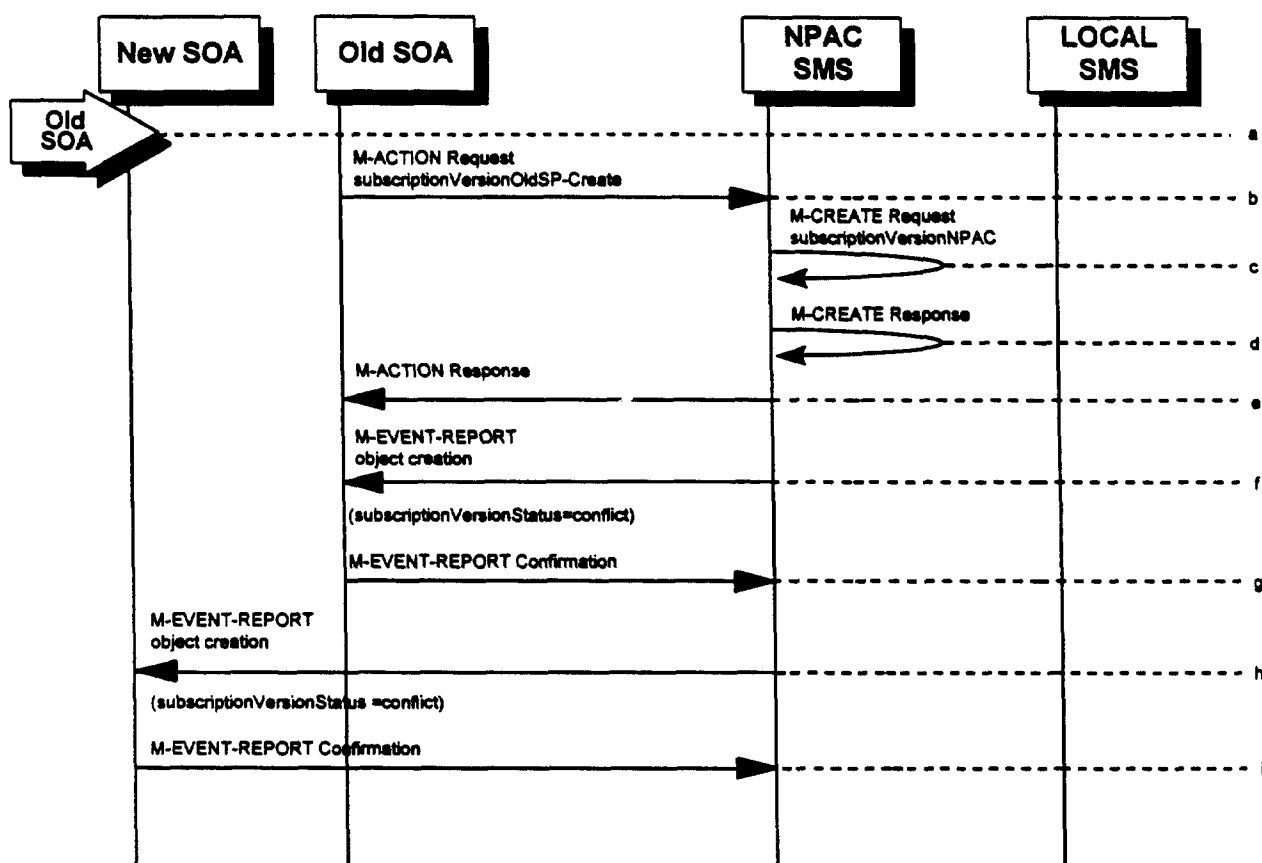
a. Action is taken by the old service provider to set a subscription version to conflict using the subscriptionVersionOldSP-Create action.

b. The old service provider SOA sends M-ACTION subscriptionVersionOldSP-Create to the NPAC SMS lnpSubscriptions object to create a new subscriptionVersionNPAC with the status of "conflict".

The old service provider SOA specifies the following valid attributes:

subscriptionTN or valid subscriptionVersionTN-Range
subscriptionNewCurrentSP
subscriptionOldSP
subscriptionOldSP-DueDate
subscriptionOldSP-Authorization
subscriptionLNPType

In this case, the subscriptionOldSP-Authorization is set to **NO**.

c. NPAC SMS issues M-CREATE to create the subscriptionVersionNPAC with a status of "conflict" and sets all the other attribute values from the subscriptionVersionOldSP-Create action.

d. NPAC SMS issues M-CREATE response.

e. NPAC SMS returns M-ACTION reply. This either reflects a success or failure and reasons for the failure.

f. If the action was successful, the NPAC SMS issues the M-EVENT-REPORT to the old service provider SOA notifying them of the object creation.

g. The old service provider SOA confirms the M-EVENT-REPORT.

h. If the action was successful, the NPAC SMS issues the M-EVENT-REPORT to the new service provider SOA notifying them of the object creation.

i. The new service provider SOA confirms the M-EVENT-REPORT.

6.5.6. SubscriptionVersion Query

This scenario shows subscriptionVersion query from service provider systems to the NPAC SMS.
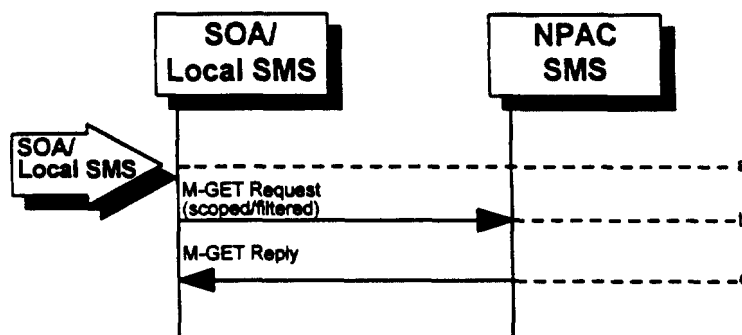
a. Action is taken by either a service provider SOA or Local SMS for retrieving one or more versions of a subscription.

b. The service provider SOA or Local SMS issues a scoped filtered M-GET from the lnpSubscriptions object to retrieve a specific version for a subscription version TN or can request all subscription versions. However, the service provider SOA is limited by a scope and filter in their search capabilities. The filter will currently support all the attributes on the subscriptionVersionNPAC.

c. The NPAC SMS replies with the requested subscriptionVersion data if the requested number of records is less than or equal to "Max Subscriber-Query" specified in the NPAC SMS. Otherwise a complexityresource-Limitation eError will be returned.

The query return data includes:

> subscriptionTN
> subscriptionLRN
> subscriptionNewCurrentSP
> subscriptionOldSP
> subscriptionNewSP-DueDate
> subscriptionNewSP-CreationTimeStamp
> subscriptionOldSP-DueDate
> subscriptionOldSP-Authorization
> subscriptionOldSP-AuthorizationTimeStamp
> subscriptionActivationTimeStamp
> subscriptionBroadcastTimeStamp
> subscriptionConflictTimeStamp
> subscriptionCustomerDisconnectDate
> subscriptionDisconnectCompleteTimeStamp
> subscriptionEffectiveReleaseDate
> subscriptionVersionStatus
> subscriptionCLASS-DPC
> subscriptionCLASS-SSN
> subscriptionLIDB-DPC
> subscriptionLIDB-SSN
> subscriptionCNAM-DPC
> subscriptionCNAM-SSN
> subscriptionISVM-DPC

```
          "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2 : 1992":objectCreation
              accessControlParameter;
    ;
```

subscriptionVersionNPAC-Definition BEHAVIOUR
    DEFINED AS !
        The subscriptionVersionNPAC class is the managed object
        that represents a subscription version on the NPAC SMS.
    !;

subscriptionVersionNPAC-Behavior BEHAVIOUR
    DEFINED AS !
        NPAC SMS Managed Object for the SOA to NPAC SMS and the Local SMS
        to NPAC SMS interface.

        A Local SMS can M-GET any subscriptionVersionNPAC objects
        from the NPAC SMS via the Local SMS Interface (Data Download
        Association Function).

        A Service Provider SOA can M-GET any subscriptionVersionNPAC
        objects from the NPAC SMS via the SOA Interface (SOA Management
        Association Function).

        If a Service Provider SOA or Local SMS does a scoped filtered
        M-GET for subscription versions, this request will only be
        successful if a the number of records to be returned is less
        than or equal to the NPAC SMS tunable parameter,
        "Max Subscriber Query", in the Service Data table.

        When the status of an object is changed to "cancel-pending",
        subscriptionPreCancellationStatus is first set to the current status.

        The subscriptionCreationTimeStamp is set to the current system
        time when the object is created.

        When the subscription version is modified for any reason, the
        subscriptionModifiedTimeStamp is updated with the current system time.

        When the subscription version is broadcast to Local SMSs via
        the NPAC to Local SMS interface, the subscriptionBroadcastTimeStamp
        is updated with the current system time.

        When the subscription version has its version status set to old,
        the subscriptionOldTimeStamp is updated with the current system
        time.

        When the subscription version has its version status set to cancel,
        the subscriptionCancellationTimeStamp is updated with the
        current system time.

        When the subscription version has its version status set to
        conflict, the subscriptionConflictTimeStamp is updated with the
        current system time.

        When the subscription version is disconnected and the version
        status is set to old, the subscriptionDisconnectCompleteTimeStamp
        is updated with the current system time.

        When the subscription version status is set to disconnect pending
        the subscriptionEffectiveReleaseDate is set to the date the
        disconnect should be broadcast.

        When the subscription version in a cancel-pending
        state is acknowledged by an old service provider SOA, the
        subscriptionOldSP-CancellationTimeStamp is updated with the
        current system time.

When the subscription version in a cancel-pending
state is acknowledged by a new service provider SOA, the
subscriptionNewSP-CancellationTimeStamp is updated with the
current system time.

When the subscription version in a conflict state is removed
from conflict by the old service provider SOA, the
subscriptionOldSP-ConflictResolutionTimeStamp is updated with the
current system time.

When the subscription version in a conflict state is removed
from conflict by the new service provider SOA, the
subscriptionNewSP-ConflictResolutionTimeStamp is updated with
the current system time.

When the subscription version status is failed or partially-failed,
the subscriptionFailed-SP-List is populated with a list of the
failed service providers.

The Service Provider SOA can M-GET and M-SET subscriptionVersionNPAC
objects via the SOA to NPAC SMS interface (SOA Management Association
Function). Rules for M-SET are described below.

For M-GET requests, the filter will support all attributes for
a specified ported TN.

Any service provider SOA can view any subscription version for any
ported TN (SOA Management Association Function).

Subscription versions are created on the NPAC SMS via actions over
the SOA to NPAC SMS interface to the lnpSubscriptions object (SOA
Management Association Function). New service provider SOAs must use
the subscriptionVersionNewSP-Create action and old service
provider SOAs must use the subscriptionVersionOldSP-Create action.
Creates can only be performed provided there is only one
currently active subscription version for the TN.

subscriptionPortingToOriginal-SPSwitch can only be specified as
TRUE for a TN that is currently ported and is being ported back
to the original service provider. If the value of
subscriptionPortingToOriginal-SPSwitch is TRUE, the LRN and GTT data
should not be specified. This data is not specified because
when the activate occurs for the subscription version, the Local
SMS will receive requests to delete the old subscription version
routing data in their networks and they will not receive any
new network routing data for the subscription. Concurrence from the
old service provider is required.

If the port of the subscription version is an intra-service
provider port, the new service provider SOA can use the
subscriptionVersionNewSP-Create action specifying the old service
provider equal to the new service provider. In this case, the
old service provider create action is not required and processing
proceeds after a valid pending version is created in the same
manner as it does for inter-service provider porting.

Once a version has been created that passes validation, the
subscriptionVersionNPAC object subscriptionVersionStatus will
be set to pending and an object creation notification will be sent
to both old and new service provider SOAs. If a version
previously existed, attribute value change notifications will be
sent to both old and new service provider SOAs.

If there is a pending version that does not have concurrence during
the "Service Provider Concurrence Window" specified in the Service
Data table, a subscriptionVersionNoConcurrence notification will be
sent to the service provider SOA that has not responded. The

subscriptionVersionStatus will be set to cancel-pending if the new
service provider SOA has not responded or to conflict if the old
service provider SOA has not responded after the "Service Provider
Concurrence Failure Window" specified in the Service Data table.
An attribute value change will be sent to the service provider
SOA that sent the original create request.

The Service Provider SOA can M-SET attributes associated with
pending, or conflict subscription versions (SOA Management Association
Function).

Attempts to modify an active, sending, failed, canceled, cancel-pending,
disconnect-pending or old version using M-SET will result in
an access denied error.

Modification of an active subscription can only be done by the
current/new service provider SOA using the subscriptionVersionModify
action.

The modify action can be used by both old and new service
provider SOAs to update pending or conflict subscription versions.

Old service provider SOAs can only modify the following attributes:

subscriptionOldSP-DueDate
subscriptionOldSP-Authorization
subscriptionStatusChangeCauseCode

New service provider SOAs can only modify the following attributes:

subscriptionLRN
subscriptionNewSP-DueDate
subscriptionCLASS-DPC
subscriptionCLASS-SSN
subscriptionLIDB-DPC
subscriptionLIDB-SSN
subscriptionCNAM-DPC
subscriptionCNAM-SSN
subscriptionISVM-DPC
subscriptionISVM-SSN
subscriptionEndUserLocationValue
subscriptionEndUserLocationType
subscriptionBillingId

Validation will be done for both old and new service provider
data that is specified on an M-SET. If validation fails, no
changes will be made and a processing failure will be returned.
If the version passes validation, the version status will be
set to pending. An error message will be returned to the service
provider if the status is not pending when they attempt to change
the version status to cancel-pending.

Once a pending version has been created, the new service provider
can activate the subscription version if authorization for the port
has been received by the old service provider within
the "Service Provider Concurrence Cancellation Window".

Once the version is activated, the version status is set to
sending, the broadcast time stamp is updated, and creates
are sent to the Local SMSs.

If the create requests are successful for all Local SMSs, the
version status will be marked as active and the previously active
subscription version will have its version status set to old.

If create requests fail for a subscription version after the
retry periods have expired, the version status will be set

to failed or partially-failed based on if the download failed
in all or some of the Local SMSs respectively.

A status version attribute value change will be sent to both old
and new service providers when the subscriptionVersionStatus is
modified. If the version status is failed or partially-failed then
a list of failed service providers is provided in the
subscriptionVersionStatus notification.

A subscription version can be put into conflict by either the NPAC
SMS or by the old service provider explicitly setting the
subscriptionOldSP-Authorization off or by not responding with the
subscriptionVersionOldSP-Create action.

If the old service provider explicitly sets the
subscriptionOldSP-Authorization off, the
subscriptionStatusChangeCauseCode must be given.

The old service provider can only put a subscription version into
conflict once and only by explicitly setting the
subscriptionOldSP-Authorization off.

A service provider should acknowledge the conflict resolution pending
state within a tunable time frame specified on the NPAC SMS
with a conflict resolution acknowledgement action.

A service provider should acknowledge the cancel pending
state within a tunable time frame specified on the NPAC SMS
with a cancel acknowledgement action.

If a service provider SOA fails to acknowledge the cancel pending
state, a subscriptionVersionCancellationAcknowledgeRequest is
sent to the service provider SOA. If they do not respond to this
acknowledgement in a tunable time frame specified on the NPAC SMS,
the version status will be set to conflict.

Attribute value change notifications will be sent to both service
provider SOAs when the following attribute values change for
a pending, cancel-pending, conflict or disconnect-pending subscription
versions:

subscriptionNewSP-DueDate
subscriptionNewSP-CreationTimeStamp
subscriptionOldSP-DueDate
subscriptionOldSP-Authorization
subscriptionOldSP-AuthorizationTimeStamp
subscriptionStatusChangeCauseCode
subscriptionVersionStatus

Object creation notifications will be sent to both old and
new service provider SOAs when a subscriptionVersionNPAC associated
with their Service Provider id is created. Object deletion
notifications will not be used. Objects will only be deleted by the
NPAC SMS as a result of housekeeping processing.

Prior to when the subscription version status is set to disconnect
pending or old, the subscriptionVersionDonorSP-CustomerDisconnectDate
is sent to the donor SOA informing the service provider of the
actual customer disconnect date.

!;

-- 22.0 LNP Log Record for the Subscription Version Donor Service Provider
--   Customer Disconnect Date Notification

lnpLogDonorSP-CustomerDisconnectDateRecord MANAGED OBJECT CLASS
    DERIVED FROM "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2 : 1992":eventLogRecord;

```
    CHARACTERIZED BY
        lnpLogDonorSP-CustomerDisconnectDatePkg;
    REGISTERED AS (LNP-OIDS.lnp-objectClass 22);

lnpLogDonorSP-CustomerDisconnectDatePkg PACKAGE
    BEHAVIOUR
        lnpLogDonorSP-CustomerDisconnectDateDefinition,
        lnpLogDonorSP-CustomerDisconnectDateBehavior;
    ATTRIBUTES
        subscriptionTN GET,
        subscriptionVersionId GET,
        subscriptionCustomerDisconnectDate GET,
        subscriptionEffectiveReleaseDate GET,
        accessControl GET;
    ;

lnpLogDonorSP-CustomerDisconnectDateDefinition BEHAVIOUR
    DEFINED AS !
        The lnpLogDonorSP-CustomerDisconnectDateRecord class is the managed
        object that is used to create log records for the
        subscriptionVersionDonorSP-CustomerDisconnectDate Notification.
    !;

lnpLogDonorSP-CustomerDisconnectDateBehavior BEHAVIOUR
    DEFINED AS !
        This log record can be used by any CME wanting to log the
        subscriptionVersionDonorSP-CustomerDisconnectDate Notification.
    !;

-- 23.0 LNP Log Record for the Subscription Version Local SMS Action Results
--      Notification

lnpLogLocalSMS-ActionResultsRecord MANAGED OBJECT CLASS
    DERIVED FROM "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2 : 1992":eventLogRecord;
    CHARACTERIZED BY
        lnpLogLocalSMS-ActionResultsPkg;
    REGISTERED AS (LNP-OIDS.lnp-objectClass 23);

lnpLogLocalSMS-ActionResultsPkg PACKAGE
    BEHAVIOUR
        lnpLogLocalSMS-ActionResultsDefinition,
        lnpLogLocalSMS-ActionResultsBehavior;
    ATTRIBUTES
        actionId GET,
        actionResultsStatus GET,
        failedTN-List GET,
        resultsCompletionTime GET,
        accessControl GET;
    ;

lnpLogLocalSMS-ActionResultsDefinition BEHAVIOUR
    DEFINED AS !
        The lnpLogLocalSMS-ActionResultsRecord class is the managed
        object that is used to create log records for the
        subscriptionVersionLocalSMS-ActionResults Notification.
    !;

lnpLogLocalSMS-ActionResultsBehavior BEHAVIOUR
    DEFINED AS !
        This log record can be used by any CME wanting to log the
        subscriptionVersionLocalSMS-ActionResults Notification.
    !;

-- 24.0 LNP Log Record for the Subscription Version New NPA-NXX Notification

lnpLogNewNPA-NXXRecord MANAGED OBJECT CLASS
    DERIVED FROM "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2 : 1992":eventLogRecord;
```

```
    CHARACTERIZED BY
        lnpLogNewNPA-NXXPkg;
    REGISTERED AS {LNP-OIDS.lnp-objectClass 24};

lnpLogNewNPA-NXXPkg PACKAGE
    BEHAVIOUR
        lnpLogNewNPA-NXXDefinition,
        lnpLogNewNPA-NXXBehavior;
    ATTRIBUTES
        serviceProvNPA-NXX-ID GET,
        serviceProvNPA-NXX-Value GET,
        serviceProvNPA-NXX-EffectiveTimeStamp GET,
        serviceProvID GET,
        accessControl GET;
    ;

lnpLogNewNPA-NXXDefinition BEHAVIOUR
    DEFINED AS !
        The lnpLogNewNPA-NXX class is the managed
        object that is used to create log records for the
        subscriptionVersionNewNPA-NXX Notification.
    !;

lnpLogNewNPA-NXXBehavior BEHAVIOUR
    DEFINED AS !
        This log record can be used by any CME wanting to log the
        subscriptionVersionNewNPA-NXX Notification.
    !;

-- 25.0 LNP Service Provider NPA-NXX Managed Object Class

lsmsFilterNPA-NXX MANAGED OBJECT CLASS
    DERIVED FROM "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2 : 1992":top;
    CHARACTERIZED BY
        lsmsFilterNPA-NXX-Pkg;
    REGISTERED AS {LNP-OIDS.lnp-objectClass 25};

lsmsFilterNPA-NXX-Pkg PACKAGE
    BEHAVIOUR
        lsmsFilterNPA-NXX-Definition,
        lsmsFilterNPA-NXX-Behavior;
    ATTRIBUTES
        lsmsFilterNPA-NXX-ID GET,
        lsmsFilterNPA-NXX-Value GET;
    ;

lsmsFilterNPA-NXX-Definition BEHAVIOUR
    DEFINED AS !
        The lsmsFilterNPA-NXX class is the managed object
        used to identify the NPA-NXX values for which a service provider
        does not want to be informed of subscription version broadcasts.
    !;

lsmsFilterNPA-NXX-Behavior BEHAVIOUR
    DEFINED AS !
        NPAC SMS Managed Object used for the Local SMS to NPAC SMS interface.

        All attributes are read only. Once created, the lsmsFilterNPA-NXX
        object can only be deleted via the Local SMS interface.  The
        lsmsFilterNPA-NXX-ID is specified by the NPAC SMS.

        The Local SMS can M-DELETE, M-CREATE and M-GET any
        lsmsFilterNPA-NXX object on the NPAC SMS.  (LSMS Network Data
        Association Function).

    !;
```

-- 26.0 LNP Log Record for the Subscription Version Final Concurrence
-- Timer Expiration

```
lnpLogOldSPFinalConcurrenceWindowExpirationRecord MANAGED OBJECT CLASS
    DERIVED FROM "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2 : 1992":eventLogRecord;
    CHARACTERIZED BY
        lnpLogOldSPFinalConcurrenceWindowExpirationPkg;
    REGISTERED AS {LNP-OIDS.lnp-objectClass 5};

lnpLogOldSPFinalConcurrenceWindowExpirationPkg PACKAGE
    BEHAVIOUR
        lnpLogOldSPFinalConcurrenceWindowExpirationDefinition,
        lnpLogOldSPFinalConcurrenceWindowExpirationBehavior;
    ATTRIBUTES
        subscriptionTN GET,
        subscriptionVersionId GET,
        accessControl GET;
    ;

lnpLogOldSPFinalConcurrenceWindowExpirationDefinition BEHAVIOUR
    DEFINED AS !
        The lnpLogOldSPFinalConcurrenceWindowExpirationRecord class is
        the managed object that is used to create log records for the
        subscriptionVersionOldSPFinalConcurrenceWindowExpiration
        Notification.
    !;

lnpLogOldSPFinalConcurrenceWindowExpirationBehavior BEHAVIOUR
    DEFINED AS !
        This log record can be used by any CME wanting to log the
        subscriptionVersionOldSPFinalConcurrenceWindowExpiration
        Notification.
    !;

--
```

7.3.   -- Name Binding Definitions

```
--

-- 1.0 LNP Audits Managed Object Name Bindings

lnpAudits-lnpNPAC-SMS NAME BINDING
    SUBORDINATE OBJECT CLASS lnpAudits AND SUBCLASSES;
    NAMED BY
        SUPERIOR OBJECT CLASS lnpNPAC-SMS AND SUBCLASSES;
    WITH ATTRIBUTE lnpAuditsName;
    -- Note: Create through interface is not supported.
    -- Note: Delete through interface is not supported.
    REGISTERED AS {LNP-OIDS.lnp-nameBinding 1};

lnpAudits-lnpLocalSMS NAME BINDING
    SUBORDINATE OBJECT CLASS lnpAudits AND SUBCLASSES;
    NAMED BY
        SUPERIOR OBJECT CLASS lnpLocalSMS AND SUBCLASSES;
    WITH ATTRIBUTE lnpAuditsName;
    -- Note: Create through interface is not supported.
    -- Note: Delete through interface is not supported.
    REGISTERED AS {LNP-OIDS.lnp-nameBinding 2};

-- 2.0 LNP Local SMS Managed Object Name Bindings

lnpLocalSMS-root NAME BINDING
    SUBORDINATE OBJECT CLASS lnpLocalSMS AND SUBCLASSES;
    NAMED BY
        SUPERIOR OBJECT CLASS "CCITT Rec. X.660 (1992) | ISO/IEC 9834-1 : 1992":root;
    WITH ATTRIBUTE lnpLocal-SMS-Name;
```

```
        -- Note: Create through interface is not supported.
        -- Note: Delete through interface is not supported.
        REGISTERED AS {LNP-OIDS.lnp-nameBinding 3};

-- 3.0 LNP Network Managed Object Name Bindings

lnpNetwork-lnpNPAC-SMS NAME BINDING
        SUBORDINATE OBJECT CLASS lnpNetwork AND SUBCLASSES;
        NAMED BY
            SUPERIOR OBJECT CLASS lnpNPAC-SMS AND SUBCLASSES;
        WITH ATTRIBUTE lnpNetworkName;
        -- Note: Create through interface is not supported.
        -- Note: Delete through interface is not supported.
        REGISTERED AS {LNP-OIDS.lnp-nameBinding 4};

lnpNetwork-lnpLocalSMS NAME BINDING
        SUBORDINATE OBJECT CLASS lnpNetwork AND SUBCLASSES;
        NAMED BY
            SUPERIOR OBJECT CLASS lnpLocalSMS AND SUBCLASSES;
        WITH ATTRIBUTE lnpNetworkName;
        -- Note: Create through interface is not supported.
        -- Note: Delete through interface is not supported.
        REGISTERED AS {LNP-OIDS.lnp-nameBinding 5};

-- 4.0 LNP NPAC SMS Managed Object Name Bindings

lnpNPAC-SMS-root NAME BINDING
        SUBORDINATE OBJECT CLASS lnpNPAC-SMS AND SUBCLASSES;
        NAMED BY
            SUPERIOR OBJECT CLASS "CCITT Rec. X.660 (1992) | ISO/IEC 9834-1 : 1992":root;
        WITH ATTRIBUTE lnpNPAC-SMS-Name;
        -- Note: Create through interface is not supported.
        -- Note: Delete through interface is not supported.
        REGISTERED AS {LNP-OIDS.lnp-nameBinding 6};

-- 5.0 LNP Service Providers Managed Object Name Bindings

lnpServiceProvs-lnpNPAC-SMS NAME BINDING
        SUBORDINATE OBJECT CLASS lnpServiceProvs AND SUBCLASSES;
        NAMED BY
            SUPERIOR OBJECT CLASS lnpNPAC-SMS AND SUBCLASSES;
        WITH ATTRIBUTE lnpServiceProvsName;
        -- Note: Create through interface is not supported.
        -- Note: Delete through interface is not supported.
        REGISTERED AS {LNP-OIDS.lnp-nameBinding 7};

-- 6.0 LNP Subscriptions Managed Object Class Name Bindings

lnpSubscriptions-lnpNPAC-SMS NAME BINDING
        SUBORDINATE OBJECT CLASS lnpSubscriptions AND SUBCLASSES;
        NAMED BY
            SUPERIOR OBJECT CLASS lnpNPAC-SMS AND SUBCLASSES;
        WITH ATTRIBUTE lnpSubscriptionsName;
        -- Note: Create through interface is not supported.
        -- Note: Delete through interface is not supported.
        REGISTERED AS {LNP-OIDS.lnp-nameBinding 8};

lnpSubscriptions-lnpLocalSMS NAME BINDING
        SUBORDINATE OBJECT CLASS lnpSubscriptions AND SUBCLASSES;
        NAMED BY
            SUPERIOR OBJECT CLASS lnpLocalSMS AND SUBCLASSES;
        WITH ATTRIBUTE lnpSubscriptionsName;
        -- Note: Create through interface is not supported.
        -- Note: Delete through interface is not supported.
        REGISTERED AS {LNP-OIDS.lnp-nameBinding 9};

-- 7.0 LNP Service Provider Managed Object Class Name Bindings
```

```
serviceProv-lnpServiceProvs NAME BINDING
    SUBORDINATE OBJECT CLASS serviceProv AND SUBCLASSES;
    NAMED BY
        SUPERIOR OBJECT CLASS lnpServiceProvs AND SUBCLASSES;
    WITH ATTRIBUTE serviceProvID;
    CREATE;
    DELETE ONLY-IF-NO-CONTAINED-OBJECTS;
    REGISTERED AS {LNP-OIDS.lnp-nameBinding 10};

-- 8.0 LNP Service Provider LRN Managed Object Class Name Bindings

serviceProvLRN-serviceProvNetwork NAME BINDING
    SUBORDINATE OBJECT CLASS serviceProvLRN AND SUBCLASSES;
    NAMED BY
        SUPERIOR OBJECT CLASS serviceProvNetwork AND SUBCLASSES;
    WITH ATTRIBUTE serviceProvLRN-ID;
    CREATE WITH-AUTOMATIC-INSTANCE-NAMING;
    DELETE ONLY-IF-NO-CONTAINED-OBJECTS;
    REGISTERED AS {LNP-OIDS.lnp-nameBinding 11};

-- 9.0 LNP Service Provider Network Managed Object Class Name Bindings

serviceProvNetwork-lnpNetwork NAME BINDING
    SUBORDINATE OBJECT CLASS serviceProvNetwork AND SUBCLASSES;
    NAMED BY
        SUPERIOR OBJECT CLASS lnpNetwork AND SUBCLASSES;
    WITH ATTRIBUTE serviceProvID;
    CREATE;
    DELETE ONLY-IF-NO-CONTAINED-OBJECTS;
    REGISTERED AS {LNP-OIDS.lnp-nameBinding 12};

-- 10.0 LNP Service Provider NPA-NXX Managed Object Class Name Bindings

serviceProvNPA-NXX-serviceProvNetwork NAME BINDING
    SUBORDINATE OBJECT CLASS serviceProvNPA-NXX AND SUBCLASSES;
    NAMED BY
        SUPERIOR OBJECT CLASS serviceProvNetwork AND SUBCLASSES;
    WITH ATTRIBUTE serviceProvNPA-NXX-ID;
    CREATE WITH-AUTOMATIC-INSTANCE-NAMING;
    DELETE ONLY-IF-NO-CONTAINED-OBJECTS;
    REGISTERED AS {LNP-OIDS.lnp-nameBinding 13};


-- 11.0 LNP Subscription Audit for the NPAC SMS Managed Object

subscriptionAudit-lnpAudits NAME BINDING
    SUBORDINATE OBJECT CLASS subscriptionAudit AND SUBCLASSES;
    NAMED BY
        SUPERIOR OBJECT CLASS lnpAudits AND SUBCLASSES;
    WITH ATTRIBUTE subscriptionAuditId;
    CREATE WITH-AUTOMATIC-INSTANCE-NAMING;
    DELETE ONLY-IF-NO-CONTAINED-OBJECTS;
    REGISTERED AS {LNP-OIDS.lnp-nameBinding 14};

-- 12.0 LNP Subscription Version Managed Object Class

subscriptionVersion-lnpSubscriptions NAME BINDING
    SUBORDINATE OBJECT CLASS subscriptionVersion AND SUBCLASSES;
    NAMED BY
        SUPERIOR OBJECT CLASS lnpSubscriptions AND SUBCLASSES;
    WITH ATTRIBUTE subscriptionVersionId;
    CREATE WITH-AUTOMATIC-INSTANCE-NAMING;
    DELETE ONLY-IF-NO-CONTAINED-OBJECTS;
    REGISTERED AS {LNP-OIDS.lnp-nameBinding 15};


-- 13.0 LNP Service Provider LSMS Filter NPA-NXX Managed Object Class
```